Reliability of Marine Structures Program

WAVEMAKER 2.0:

SIMULATION AND IDENTIFICATION OF SECOND-ORDER RANDOM WAVES

Alok K. Jha
Steven R. Winterstein

Civil Engineering Department, Stanford University

Supported by
Office of Naval Research · Stanford RMS Program

June 1996
Report No. RMS–22

Department of CIVIL ENGINEERING
STANFORD UNIVERSITY

WAVEMAKER 2.0:

SIMULATION AND IDENTIFICATION OF SECOND-ORDER RANDOM WAVES

Alok K. Jha
Steven R. Winterstein

Civil Engineering Department, Stanford University

Supported by
Office of Naval Research · Stanford RMS Program

June 1996
Report No. RMS–22

Acknowledgments

These simulation and identification analysis capabilities have been developed by the first author during the course of his ongoing Ph.D. studies. Early portions of these studies were supported principally by the industry sponsors of the Reliability of Marine Structures Program of Stanford University.

Most recent developments have been funded by the Office of Naval Research, grant N00014-95-1-0366, under the supervision of Dr. Peter Majumdar, and grant N00014-96-1-0641, under the supervision of Dr. Roshdy S. Barsoum. These developments have included spatial modeling of nonlinear waves, for application to stochastic ship load and response calculation, and identification analysis capabilities, for application to offshore structural analysis.

We gratefully acknowledge these sources of support.

Table of Contents

List of Figures

Abstract

WAVEMAKER is a FORTRAN subroutine to simulate random non-Gaussian ocean wave histories. It generates a first-order (Gaussian) wave process with an arbitrary power spectrum, and applies nonlinear corrections based on second-order hydrodynamics. Inputs to the routine include the first-order spectrum, the water depth, and a set of locations in the along-wave direction at which wave elevation histories are desired. It may thus provide useful input to estimate loads on spatially distributed ocean structures and ships.

The WAVEMAKER package also includes a separate driver program, which facilitates input/output and generates several analytical spectral models. Its input is specified in command-line format, similar to that of the TF-POP program for hydrodynamic post-processing also developed in the Stanford RMS program. An example problem is included to demonstrate the use of WAVEMAKER and its driver.

In terms of methodology, WAVEMAKER first uses standard frequency domain methods to generate first-order Gaussian histories at each location. For each of these, WAVEMAKER then evaluates the full set of second-order corrections according to hydrodynamic theory. Thus the first-order wave process, with N components at frequencies ω_n, gives rise to a total of N^2 corrections, spread over all sum frequencies $\omega_n + \omega_m$, and to another N^2 corrections over all difference frequencies $\omega_n - \omega_m$.

WAVEMAKER also includes the ability to identify the underlying first-order Gaussian history from a given observed time history. This feature is particularly attractive for use in situations where the second-order nonlinearity in the waves is built-in into the structural response calculations. To avoid double-counting therefore, the input waves should be filtered to remove any second-order nonlinearity. WAVEMAKER takes in an input wave history and identifies its first- and second-order wave components. This identification, an inverse feature to simulation, is based on a Newton-Raphson scheme to solve N simultaneous nonlinear equations to identify the first-order waves which, when run through the second-order wave predictor, matches the observed waves.

Chapter 1

Introduction to WAVEMAKER 2.0

This release of WAVEMAKER software incorporates a major recent development achieved at the Reliability of Marine Structures Program. This is the ability to successfully identify the underlying first-order wave components for given target observed wave histories. WAVEMAKER 2.0 is fully backward compatible, that is, results from a run of earlier versions of WAVEMAKER can be exactly obtained with this new release. The input files to earlier versions can be directly used in this new version. A version history of WAVEMAKER follows:

- **Version 1.0**: Released in April 1995 and documented in Jha and Winterstein, 1995, Report RMS–17, contains simulation capabilities for second-order random waves.
- **Version 1.1**: Released in August 1995, includes modification of temporary intermediate output file to use less disk space and to reduce program execution time by approximately 50%. Additionally, a DOS executable of WAVEMAKER was included in this release.
- **Version 2.0**: Released June 1996, includes identification capabilities so that underlying first-order wave history can be retrieved from an observed wave history.

In this report, Chapter 2 includes a documentation of the simulation capabilities and is largely taken from Report RMS–17. Chapter 3 documents the newly developed identification analysis capabilities, and the appendices include sample input and output files for the simulation and identification examples presented in this report.

Chapter 2

Simulation of Second-Order Random Waves

2.1 Introduction

It is common in many ocean engineering problems to seek to simulate a time trace of the wave elevation process, $\eta(t)$, at one or more locations in the along-wave direction. It is most typical to use a *Gaussian* model of $\eta(t)$ for this simulation, which is consistent with linear wave theory. This is due primarily to the ease of simulating Gaussian processes, e.g. with FFT (Fast Fourier Transform) methods for an arbitrary wave spectrum (e.g., Borgman, 1969).

We seek here to demonstrate and facilitate a similar frequency-domain simulation capability for *nonlinear* random waves at a set of spatial locations (e.g., Figure 2.1). These simulations split the wave elevation into a random first-order (linear) wave history, $\eta_1(t)$, and a corresponding nonlinear history $\eta_2(t)$ which includes second-order corrections. FFT techniques are used to generate $\eta_1(t)$ with an arbitrary (first-order) wave spectrum, $S_{\eta_1}(\omega)$. Physical principles are used to generate $\eta_2(t)$ from $\eta_1(t)$, based on second-order perturbation analysis of the underlying nonlinear hydrodynamic problem. Thus if the first-order wave process has N components, at frequencies ω_n, $\eta_2(t)$ includes N^2 second-order corrections, spread over all sum frequencies $\omega_n + \omega_m$, and another N^2 corrections over all difference frequencies $\omega_n - \omega_m$.

Note that these second-order wave models are not novel; they date back at least to the early 1960s (e.g., Longuet-Higgins, 1963). More novel, however, is their recent confirmation with respect to various statistics of field measurements (Marthinsen and Winterstein, 1992; Vinje and Haver, 1994) and wave tank studies of still more severe

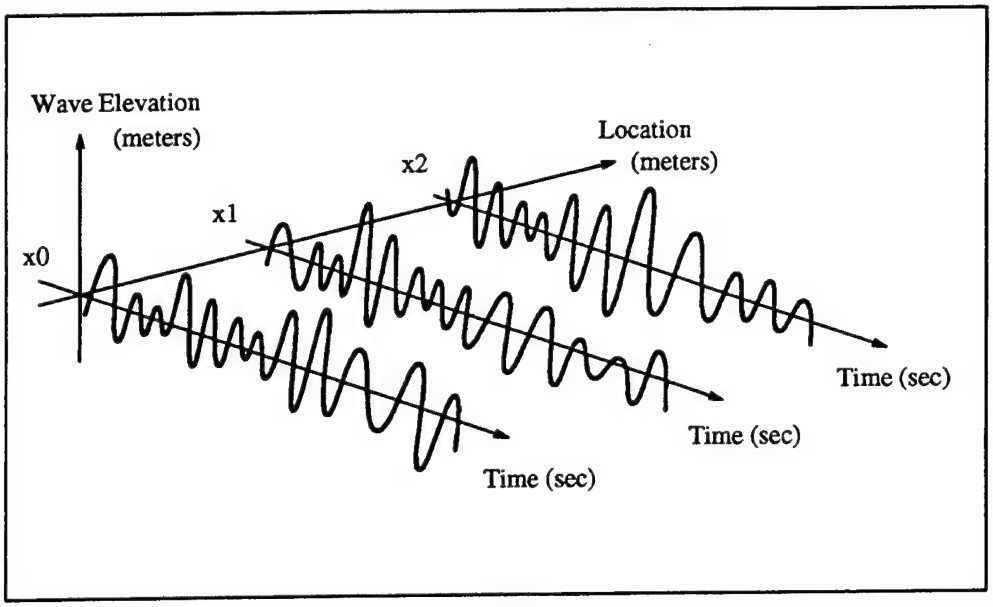

Figure 2.1: Simulated wave time histories at specified spatial locations

seas (Winterstein and Jha, 1995; preprint included in Appendix C). Such second-order wave models also form the basis of state-of-the-art nonlinear diffraction analysis of floating structures (e.g., SWIM, 1995; WAMIT, 1995). Note also that the model of $\eta_2(t)$ used here varies explicitly with water depth, as predicted by second-order theory, to reflect increasing nonlinearity as we proceed to shallower water depths.

2.2 Methodology

2.2.1 Underlying Theory and Assumptions

We first consider $\eta_1(t)$, the first-order wave elevation, at a specific reference location (say x=0). For either frequency-domain analysis or time-domain simulation, it is convenient to write $\eta_1(t)$ as a discrete Fourier sum over positive frequencies ω_k:

$$\eta_1(t) = \sum_{k=1}^{N} A_k \cos(\omega_k t + \theta_k) = \mathrm{Re} \sum_{k=1}^{N} A_k e^{i(\omega_k t + \theta_k)} \tag{2.1}$$

To randomize Eq. 2.1, the phases θ_k are taken to be uniformly distributed, mutually independent of each other and of the amplitudes A_k. Furthermore, we assign

random amplitudes A_k with Rayleigh distributions, and mean-square value

$$E[A_k^2] = 2S_\eta(\omega_k)d\omega \equiv \sigma_k^2; \quad d\omega = \omega_k - \omega_{k-1} \tag{2.2}$$

Finally, for purposes of simulation the lowest frequency interval $d\omega$ is governed by the total period T of the simulation:

$$d\omega = \frac{2\pi}{T} \tag{2.3}$$

Together, Eqs. 2.1–2.2 ensure that each of the N frequency components in Eq. 2.1 is itself Gaussian. We also caution against the common use of choosing *deterministic* amplitudes, A_k=σ_k, particularly when interest lies in preserving higher moments of $\eta_1(t)$—or, similarly, the rms of second-order waves, loads, and responses. Use of deterministic amplitudes can give unconservative estimates; e.g., second-order rms values that are on average too small (Ude, 1994).

The resulting second-order wave at this elevation, $\eta_2(t)$, is calculated from $\eta_1(t)$ as

$$\eta_2(t) = \eta_1(t) + \Delta\eta_2(t) \tag{2.4}$$

in which $\Delta\eta_2(t)$ includes second-order corrections at sums and differences of all wave frequencies:

$$\Delta\eta_2(t) = q\mathrm{Re}\sum_{m=1}^{N}\sum_{n=1}^{N} A_m A_n [H_{mn}^- e^{i[(\omega_m-\omega_n)t+(\theta_m-\theta_n)]} + H_{mn}^+ e^{i[(\omega_m+\omega_n)t+(\theta_m+\theta_n)]}] \tag{2.5}$$

In general, the functions H_{mn}^- and H_{mn}^+ are known as quadratic transfer functions (QTFs), evaluated at the frequency pair ω_m, ω_n. Similar expressions arise in describing loads and responses of floating structures; in this case H_2^- and H_2^+ are calculated numerically from nonlinear diffraction analysis (e.g., WAMIT, 1995). The leading factor q is included in Eq. 2.5 to alert readers to different QTF definitions in the literature: various diffraction analyses use q=1 (WAMIT, 1995) or q=1/2 (Molin and Chen, 1990).

In predicting motions of floating structures, in view of the relevant natural periods interest commonly lies with either H_2^- (slow-drift) or H_2^+ (springing) but not both. In contrast, in the nonlinear wave problem both sum and difference frequency effects play a potentially significant role. Fortunately, unlike QTF values found numerically from numerical diffraction, closed-form expressions are available for both the sum- and difference-frequency QTFs for second-order waves (e.g., Langley, 1987; Marthinsen

and Winterstein, 1992). Including the effect of a finite water depth d, for example, the sum-frequency QTF can be written as

$$H_{mn}^{+} = \frac{\frac{gk_mk_n}{\omega_m\omega_n} - \frac{1}{2g}(\omega_m^2+\omega_n^2+\omega_m\omega_n) + \frac{g}{2}\frac{\omega_mk_n^2+\omega_mk_n^2}{\omega_m\omega_n(\omega_m+\omega_n)}}{1 - g\frac{k_m+k_n}{(\omega_m+\omega_n)^2}\tanh(k_m+k_n)d} - \frac{gk_mk_n}{2\omega_m\omega_n} + \frac{1}{2g}(\omega_m^2+\omega_n^2+\omega_m\omega_n) \quad (2.6)$$

in which the wave numbers k_n are related to the frequencies ω_n by the linear dispersion relation. Note that this QTF definition assumes q=1/2 in Eq. 2.5. This is the convention assumed in **WAVEMAKER**. The corresponding difference-frequency transfer function, H_{mn}^{-}, is found by replacing ω_n by $-\omega_n$ in Eq. 2.6.

2.2.2 Implementation

On input the simulation method requests the desired number of simulated points, *npts*, and the total duration T to be simulated. To take advantage of discrete FFT (Fast Fourier Transform) techniques, it assumes a regular spacing dt=$T/npts$ between points. Eq. 2.1 is then rewritten as

$$\eta_1(t) = \sum_{k=1}^{npts/2} A_k\cos(\omega_k t+\theta_k) = \mathrm{Re}\sum_{k=1}^{npts} X_k e^{i\omega_k t} \quad (2.7)$$

Here the X_k are complex Fourier coefficients. The lower half of these directly reflect both the random amplitude A_k and phase θ_k at frequency ω_k=$k\cdot d\omega$:

$$X_k = \frac{1}{2}A_k e^{i\theta_k}\,;\quad k = 1...npts/2 \quad (2.8)$$

The upper half are in turn taken as the complex conjugates (the symbol "*") of the lower half:

$$X_{npts-k} = X_k^*\,;\quad k = 1...npts/2 \quad (2.9)$$

This reflects that unique information is contained only the lower-half frequencies; indeed, any information in the upper half frequencies (above the Nyquist) is obscured by aliasing.

Thus, the first-order wave process is generated by assigning random A_k and θ_k, defining X_k from Eqs. 2.8–2.9, and finally taking the inverse Fourier transform to recover the discretized time history $\eta_1(t_j)$. To see this, note that since $dt\cdot d\omega$=$2\pi/npts$,

Eq. 2.7 can be evaluated at $t=t_j$ to give

$$\eta_1(t_j) = \mathrm{Re} \sum_{k=1}^{npts} X_k e^{2\pi i j k/npts} \tag{2.10}$$

This is precisely the definition of the discrete FFT.

As a minor technical issue, note that the conjugate symmetry here ensures that the "Real Part" operation in Eq. 2.1 is superfluous; i.e., no imaginary component is generated. Also, to conform with the FFT routine used the array of X_k values is shifted by one index: i.e., X_1 corresponds to the frequency zero (the steady term, defined as zero), X_2 to frequency $\omega_1=d\omega$, X_3 to frequency $\omega_2=2 \cdot d\omega$, and so forth.

The second-order correction is generated similarly. Starting with Eq. 2.5, substituting $q=1/2$, $N=npts/2$ (Eq. 2.7) and X_k from Eq. 2.8:

$$\Delta\eta_2(t) = 2\mathrm{Re} \sum_{m=1}^{npts/2} \sum_{n=1}^{npts/2} X_m X_n H_{mn}^{+} e^{i(\omega_m+\omega_n)t} + X_m X_n^{*} H_{mn}^{-} e^{i(\omega_m-\omega_n)t} \tag{2.11}$$

The leading factor reflects the product of $q=1/2$ and a net factor of 4 (since $A_n A_m$ is $4\,|X_m X_n|$). The program then seeks to rewrite both the sum and difference frequency contributions in a Fourier sum analogous to Eq. 2.10. For example, the sum-frequency is assumed of the form

$$\Delta\eta_2^{+}(t_j) = \mathrm{Re} \sum_{k=1}^{npts} Y_k e^{2\pi i j k/npts} \tag{2.12}$$

The output Fourier coefficients, Y_k, are evaluated by equating Eqs. 2.11 and 2.12. This implies a sum over all wave frequency pairs (ω_m, ω_n) in Eq. 2.11 that give rise to output sum frequency ω_k. The difference frequency Fourier coefficients are constructed in a similar way, and added on the sum frequency Y_k coefficients. Once these combined Y_k coefficients are found a (one-dimensional) inverse FFT is performed to recover the second-order time history.

2.2.3 Multiple Spatial Locations

The linear dispersion relation can be used to generalize Eq. 2.4, which generates a first-order wave at reference location $x=0$, to any other spatial location x in the along-wave direction. The linear dispersion relation is first used to find the wave number k_n associated with each frequency ω_n in Eq. 2.4. The modified first-order simulation then merely replaces $\omega_n t + \theta_n$ in Eq. 2.4 by $\omega_n t - k_n x + \theta_n$. Equivalently, the original phases θ_n are first shifted to $\theta_n - k_n x$ before applying Eq. 2.8 to define the wave Fourier amplitude X_n. These appropriately modified X_n are also used in Eq. 2.11 to find the corresponding second-order correction at this new location.

2.3 Input Specification

This section describes the various inputs required by the program and the syntax of the input to the driver routine for **WAVEMAKER**. This input is provided in the following format:

keyword *args*
where keyword is a reserved word and *args* are its arguments. A typical input file is in the following format:

```
# Typical input file: syntax description

simulate duration npts seed
depth value
psd psdtype psd_parameters
define varlimit value
define gravity value
write history filename1 filename2
write statistics filename3 filename4
location nloc
value1
value2
  ⋮
valuenloc
```

Each of these lines in a typical input file is explained below:

 # Typical input file: syntax description
Any line beginning with a "#" is treated as a comment line in the input file and is ignored by the program. Blank lines are also ignored by the program.

simulate *duration npts seed*
The keyword **simulate** indicates to the program that the following three arguments in sequence are:

- *duration*: Total desired duration (in seconds) of each of the simulated wave histories (a real number).

- *npts*: Number of points required in each of the simulated wave histories (an integer).
- *seed*: A real number (131071.0 for example) for generation of random numbers. The seed may be changed by the user in order to generate a different set of wave histories. The number should be between 1.0 and 2^{31}.

The resulting dt (time step) in the simulated histories is *duration/npts*. The *mxgrd* variable in the driver program controls the maximum number of points allowed (Maximum $npts = 2 \times mxgrd$) in a simulation. The released driver program has $mxgrd = 4096$, so that specified *npts* can to up to 8192. The user may increase or decrease *mxgrd* to suit his/her needs.

depth *value*
This line specifies the water depth at the site of interest. The keyword is **depth** and *value* is a real number indicating the water depth. The units (meters, feet, etc.) of this value should be consistent with the units of other input parameters.

psd *psdtype psd_parameters*
This line specifies the spectrum type to be used. The keyword is **psd** followed by its arguments. *psdtype* may be one of the following: jonswap, bimodal, or boxcar. If any other word is specified for *psdtype*, then it indicates to the program that an input spectrum is specified in a file whose name is same as the word specified in place of *psdtype*. More details regarding this input specification are given in the following subsection.

define varlimit *value* [optional command]
If included, this line defines a constant **varlimit** whose value is a real number (between 0.0 and 1.0) equal to *value*. A warning is issued by the simulation routine, if the estimated second-order power above Nyquist frequency is more than *value* times the first-order power. If this line is not provided in the input file, then a default value of 0.01 is assigned to **varlimit**.

define gravity *value* [optional command]
If included, this line specifies the acceleration due to gravity in consistent units. *value*, a real number, is assigned to **gravity**. If not included, a default value of 9.807 meters/sec^2 is assumed.

write history *filename1 filename2* [optional command]
If included, this line specifies the files which contain the simulated time histories at each location. The keyword is **write history**. The file *filename1* contains the underlying first-order (Gaussian) histories, and *filename2* contains the corresponding second-order wave histories. The format of the output is presented in the following section. If this line is not provided in the input file then default names of **gauss.hist** and **ngauss.hist** are assigned to the output first- and second-order histories, respectively.

write statistics *filename3 filename4* [optional command]
This line specifies the files to which the statistics (mean, standard deviation, skewness, kurtosis, minimum, and maximum value) of the simulated histories should be written. The keyword is **write statistics**. The statistics for the simulated first-order histories at each spatial location is written out in *filename3* and the statistics for the total second-order histories are written in *filename4*. If this line is not provided in the input file then default names of **gauss.stat** and **ngauss.stat** are assigned to the output files for the first- and second-order history statistics, respectively.

location *nloc*
This line specifies the number of spatial locations at which both the first- and total second-order wave histories should be simulated in time. The keyword is **location**. *nloc* (an integer) specifies the number of locations (maximum location allowed is 50). *value1*, *value2*, ..., *valuenloc* are the spatial values (real numbers) in consistent units. The number of values should be equal to *nloc*. The user is forewarned that the specification of the spatial locations should be at the end of all other inputs required.

2.3.1 Wave Spectrum Specification

The input wave spectrum can be specified in various ways: spectrum values in an input file, or spectrum type with related parameters from a default library.

psd *filename*
This specifies that the input spectrum be read from an input file named *filename*. The format of this file is two free-formatted values per line, specifying a natural frequency in radians per second and the one-sided PSD (power spectral density) ordinate for that frequency in consistent units of squared amplitude (feet^2, meters^2, etc.) per

rad/sec. Lines beginning with a "#" are regarded as comments and ignored. The input spectrum may be specified on an irregular mesh. This is internally linearly interpolated to a spectrum on a regular mesh specified by *duration* and *npts*. The spectral ordinates below the minimum frequency and above the maximum frequency specified in the irregular mesh are assumed to be zero.

Spectral models from the library are called upon using any one of the following reserved names followed by their parameters (that are real numbers):

psd jonswap H_s T_p γ
psd bimodal H_s T_p
psd boxcar σ_η $\omega_{\rm lo}$ $\omega_{\rm hi}$

The keyword **jonswap** invokes a JONSWAP spectrum parameterized by the significant wave height H_s (defined as four times the standard deviation of the wave elevation process), spectral peak period T_p (in seconds), and the peakedness factor γ.

The keyword **bimodal** invokes a spectral model proposed by Torsethaugen (Bitner-Gregerson and Haver, 1991). This subdivides the H_s–T_p scattergram into three regions, and assigns bimodal spectral shapes in several of these regions. Therefore, the only input required for this **bimodal** option is H_s and T_p.

Finally, the keyword **boxcar** invokes a simple band-limited white-noise model of the first-order wave spectrum. Its parameters are the rms σ_η, lower cutoff frequency $\omega_{\rm lo}$, and upper cutoff frequency $\omega_{\rm hi}$ of the first-order wave spectrum. As in other cases (e.g., the user-defined spectrum at various frequencies), the frequencies $\omega_{\rm lo}$, and $\omega_{\rm hi}$ here are assumed to be in units of rad/sec. Note also that non-zero values of the PSD at zero frequency are not allowed in either file input or library model selections.

2.4 Output Format

A total of four output files are produced by the driver program. Two output files contain the time histories: one for the underlying first-order wave histories and the other for the total second-order histories. The other two output files contain wave statistics: first four moments, minimum and maximum. Again, results for the first- and second-order wave histories are separated into two files.

2.4.1 Time History Output

As noted in the previous section, by default the first- and second-order histories are written to the files **gauss.hist** and **ngauss.hist**. Other choices of output filenames can be specified by the optional **write history** command. The format of this output depends on the number of spatial locations specified. If the number of locations (*nloc*) is less than or equal to 8 then the output is in **Format1** otherwise the output is in **Format2**. Both of these formats write out 3 header lines beginning with a "#" sign. These are to be treated as comment lines in the output file. In order to explain the two format styles, say that the spatial locations specified are x1, x2, x3, ..., xnloc.

Format1 outputs data in *nloc*+1 columns. The length of each column is equal to the number of points desired in each simulation. The first column contains the time increments in seconds going from 0 to T with dt = $T/npts$. Columns 2 through *nloc*+1 contain the simulated history values at the specified locations x1, x2, x3, ...xnloc, respectively. Thus, column 2 contains the wave elevation at location x1, column 3 contains wave elevation at location x2, and so on.

Format2 is for handling *nloc* greater than 8. The output begins with the time increment Ti in seconds on a line by itself. The time history values for the specified spatial locations at time Ti are written in the next line onwards, in sets of 10. So if 9 locations were specified (i.e., *nloc* = 9) then the time increment is printed on a line by itself followed by a line containing 9 time history values at that time increment. The next line contains the next time increment followed by another set of 9 values, and so on. If, on the other hand say 28 locations were specified, then a time increment is written on a line followed by 28 time history values (corresponding to 28 locations at that time increment) in the next 3 lines. The first line of the 3 lines contains 10 time history values for the first 10 locations specified. The next line contains 10 history values for locations 11 through 20 and the following line which is the third line of the set will contain only 8 history values for location 21 through 28.

2.4.2 Wave Statistics Output

The statistics of the simulated histories are also estimated by the driver program. These statistics include the mean, standard deviation, skewness, kurtosis, minimum, and maximum. As noted in the previous section, first- and second-order simulation results are written by default to the files **gauss.stat** and **ngauss.stat**, respectively. The optional command **write statistics** can alter this choice of output filenames.

The output format in both of these files begins with 2 header lines, each of which

begins with a "#" sign. The output is in seven columns. The first column specifies the spatial location. The following six columns contain statistics of the wave history at the spatial location specified in column 1. Columns 2 through 7 contain, the mean, standard deviation, skewness, kurtosis, minimum, , and maximum value in that order.

The next section presents some sample output files, to illustrate use of the **WAVEMAKER** routine.

2.5 Example

In this section, we present a sample problem (copies of input and output files are enclosed on disk). To illustrate, consider a simulation which samples the wave process at regular intervals of length dt=0.5 [sec] over a total duration of T=2048 [sec]; i.e., $npts$=4096 points. We assume here the first-order wave spectrum to be of JONSWAP form, with $H_s = 12$ [m], $T_p = 14$ [sec] and a peakedness factor $\gamma = 3.3$. We further seek to generate wave histories at 2 spatial locations: 0 and 60 [m]. Our input file for simulating waves using **WAVEMAKER** should be:

```
# Gaussian and Nongaussian Wave Input File

simulate 2048.0 4096 8123872.0
depth 70.0
psd jonswap 12.0 14.0 3.3
write history gauss.hist ngauss.hist
write statistics gauss.stat ngauss.stat
define varlimit 0.01
define gravity 9.807
location 2
0.0
60.0
```

Alternatively, if we intend to use the default definitions in the program then our input file could be (this will produce the same output as the extended version of the input file):

```
# Gaussian and Nongaussian Wave Input File
# (Alternative format)

simulate 2048.0 4096 8123872.0
depth 70.0
psd jonswap 12.0 14.0 3.3
location 2
0.0
60.0
```

The output files created are: **gauss.hist**, **ngauss.hist**, **gauss.stat**, and **ngauss.stat**. The contents of these are listed in the following table:

Output File	Contents
gauss.hist	First-Order Time History
ngauss.hist	Second-Order Time History
gauss.stat	First-Order History Statistics
ngauss.stat	Second-Order History Statistics

Each of these output files is given in the Appendix A.

Figures 2.2 and 2.3 show a comparison of the simulated histories. Figure 2.2 compares the simulated first- and second-order wave time histories at the same location (the first of the two requested, defined arbitrarily as x=0 [m]). The file **ngauss.stat** includes the estimated skewness of the second-order waves. At this location the second-order wave history has a skewness of about 0.2. This positive skewness (compared to zero skewness of Gaussian waves) indicates the systematic nonlinear effects. This also gives rise to an asymmetry between peaks and troughs; in particular, the extreme wave crest in the second-order simulation systematically exceeds the corresponding extreme trough in absolute value. This tendency may be significant for potential deck impact problems, particularly in older jacket structures with relatively low deck levels.

Figure 2.3 compares the second-order wave histories predicted by simulation at the two spatial locations, separated by 60 [m]. The phase lag at these two locations is evident in the plot. It can be seen that a crest height at 0 [m] does not necessarily imply a crest of the same height at 60 [m].

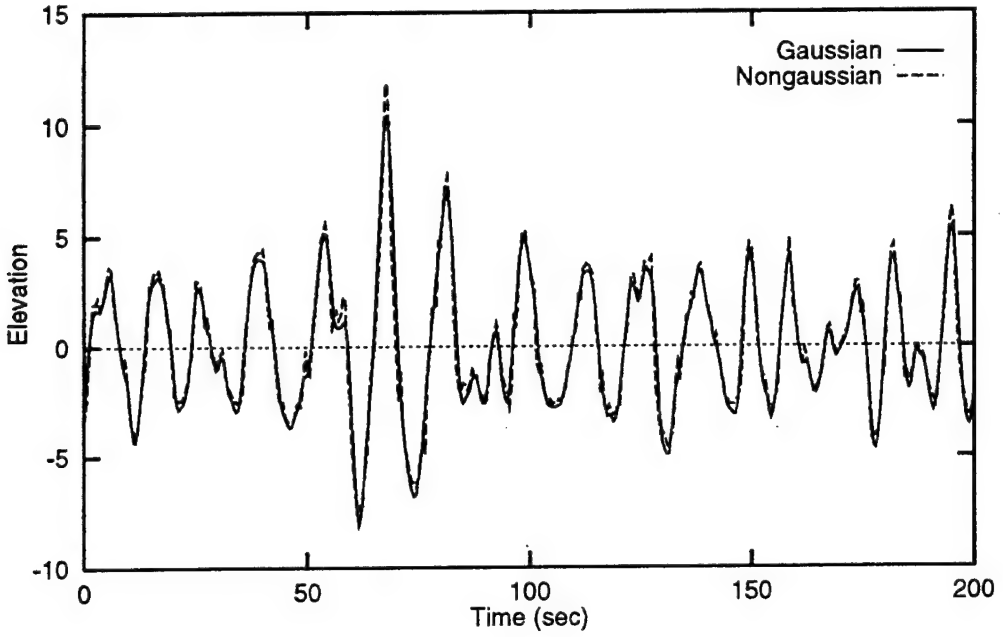

Figure 2.2: Simulated first- and second-order wave histories at location 0.0

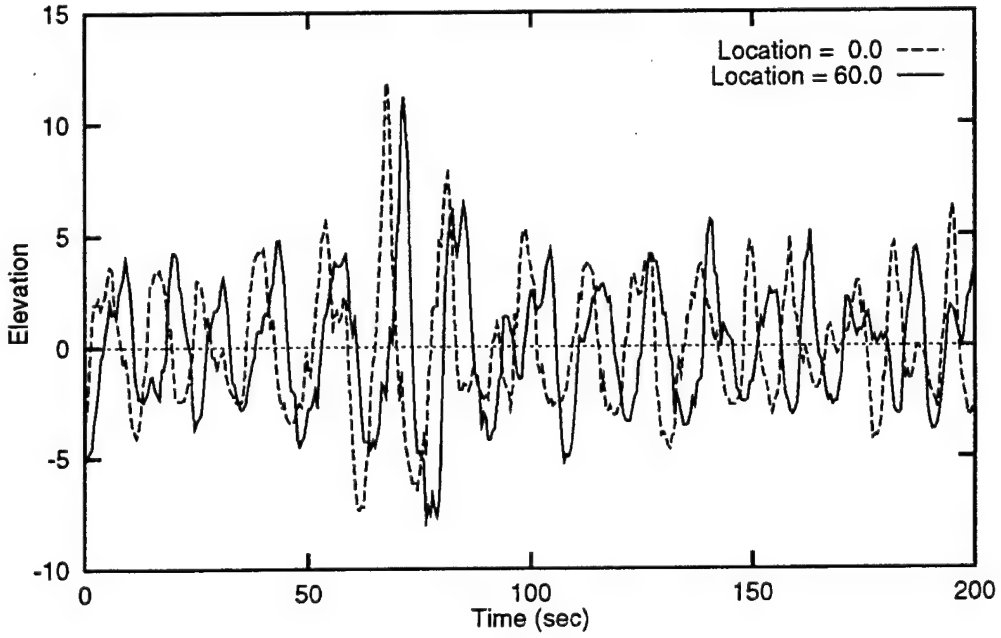

Figure 2.3: Simulated second-order wave histories at location 0.0 and 60.0

Chapter 3

Identification of First-Order Waves

3.1 Introduction

In ocean engineering practice it is common to assume the waves to be Gaussian and any nonlinearity in the waves is embedded in the structural response analysis (e.g., WAMIT, 1995). It has been shown in Winterstein and Jha, 1995 that observed time histories generally contain nonlinearities, it is thus imperative to remove any second-order effects in the incident waves so that we do not double-count these effects in the resulting response estimation. Recent studies (Ude and Winterstein, 1996) have demonstrated the impact of double-counting such second-order effects on various structural response characteristics.

The methodology to identify the underlying first-order waves is to seek the implied first-order wave history which, when run through the second-order wave predictor, yields an incident wave that agrees with the target observed history at each time point. This identification is performed using a Newton-Raphson scheme to achieve simultaneous convergence at each complex Fourier component. If the observed history has N components, we iteratively solve N simultaneous nonlinear equations to identify the first-order components.

Due to computer memory limitations, the identification of the first-order history is performed on short contiguous windows of the observed history. This window size (*mlen*) can be made equal to the observed history length in `WAVEMAKER` if the computer has sufficient RAM and swap space.

3.2 Methodology

The idea here is to identify the implied first-order history $\eta_1(t)$ (of an observed history $\eta_{\text{obs}}(t)$) which, when run through the second-order predictor, yields an incident wave that agrees with $\eta_{\text{obs}}(t)$. In the first-order wave process $\eta_1(t)$ (see Eq. 2.7), written as a Fourier sum of N frequencies,

$$\eta_1(t) = \sum_{k=1}^{N/2} A_k \cos(\omega_k t + \theta_k) = \sum_{k=1}^{N} X_k e^{i\omega_k t} \tag{3.1}$$

we need to identify only the lower half X_k components, since the upper half values are complex conjugates of the lower half. Let us denote $X_k = U_k + iV_k$, where U_k, V_k are the real amd imaginary parts of the complex Fourier component X_k, respectively.

The predicted second-order wave process (see Eq. 2.11) as evaluated from the QTFs is

$$\Delta\eta_2(t) = 2\text{Re} \sum_{m=1}^{N/2} \sum_{n=1}^{N/2} X_m X_n H_{mn}^{+} e^{i(\omega_m+\omega_n)t} + X_m X_n^{*} H_{mn}^{-} e^{i(\omega_m-\omega_n)t} \tag{3.2}$$

This may be rewritten in the form of a Fourier sum as

$$\Delta\eta_2(t) = \sum_{k=1}^{N} Y_k e^{i\omega_k t} \tag{3.3}$$

where $Y_k = Y_k^{+} + Y_k^{-}$ are the combined sum and difference frequency components. Here again, Y_k possesses conjugate symmetry so that only the lower half contains unique information. Y_k^{+} can be shown to be

$$\begin{aligned} Y_k^{+} &= \sum_{m+n,k} X_m X_n H_{mn}^{+} \\ &= \sum_{m+n,k} \left[(U_m U_n - V_m V_n) + i(V_m U_n + U_m V_n)\right] H_{mn}^{+} \end{aligned} \tag{3.4}$$

where the summation symbol indicates a double summation

$$\sum_{m+n,k} = \sum_{m=1}^{N/2} \sum_{n=1}^{N/2} \quad \text{such that } \omega_m + \omega_n = \omega_k \tag{3.5}$$

and

$$\begin{aligned} Y_k^{-} &= \sum_{m-n,k} X_m X_n^{*} H_{mn}^{-} \\ &= \sum_{m-n,k} \left[(U_m U_n + V_m V_n) + i(V_m U_n - U_m V_n)\right] H_{mn}^{-} \end{aligned} \tag{3.6}$$

where

$$\sum_{m-n,k} = \sum_{m=1}^{N/2} \sum_{n=1}^{N/2} \quad \text{such that } |\omega_m - \omega_n| = \omega_k \tag{3.7}$$

The combined predicted wave process is

$$\eta_{\text{pred}}(t) = \eta_1(t) + \Delta\eta_2(t) \tag{3.8}$$

The identification scheme strives to simultaneously match $\eta_{\text{pred}}(t)$ to the observed wave history $\eta_{\text{obs}}(t)$ at every value of t. Alternatively, we can perform the identification in the frequency domain and strive to simultaneously match the predicted Fourier components to the observed Fourier components at all frequencies.

$\eta_{\text{obs}}(t)$ can be represented in the frequency domain as

$$\eta_{\text{obs}}(t) = \sum_{k=1}^{N} Z_k e^{i\omega_k t} \tag{3.9}$$

where Z_k's also possess conjugate symmetry. If the first-order components are identified exactly, from Eq.s 3.1, 3.3 and 3.9 we will have

$$Z_k = X_k + Y_k \quad ; \quad \text{for all } k = 1 \ldots N/2 \tag{3.10}$$

Note that the upper half values can be obtained from conjugate symmetry of the lower half values. In the Newton-Raphson identification scheme we will try to simultaneously minimize $X_k + Y_k - Z_k$; for $k = 1 \ldots N/2$ to achieve convergence. Now, this scheme requires a Jacobian of $X_k + Y_k - Z_k$ with respect to the unknowns X_k–such a complex differentiation will lead to numerical discontinuities so we will minimize an equivalent real function $\sqrt{\sum_1^N f_k^2/N}$ instead, where for $k = 1 \ldots N/2$

$$\begin{aligned} f_k &= \text{Re}(X_k + Y_k - Z_k) \\ f_{k+N/2} &= \text{Im}(X_k + Y_k - Z_k) \end{aligned} \tag{3.11}$$

The identification of the lower half X_k values requires a simultaneous solution of the nonlinear equations in 3.11 such that $f_k \to 0$ for all $k = 1 \ldots . N$, or alternately $\sqrt{\sum_1^N f_k^2/N} \to 0$. We will formulate the Newton-Raphson scheme in vector form as

$$\mathbf{f} = \left[\frac{\text{Re}\mathbf{X}}{\text{Im}\mathbf{X}}\right] + \left[\frac{\text{Re}\mathbf{Y}}{\text{Im}\mathbf{Y}}\right] - \left[\frac{\text{Re}\mathbf{Z}}{\text{Im}\mathbf{Z}}\right] \tag{3.12}$$

where bold face letters denote vectors, and vectors $\mathbf{X}, \mathbf{Y}, \mathbf{Z}$ contain the complex Fourier components X_k, Y_k, Z_k, $k = 1 \ldots N/2$, respectively. Here, $\left[\frac{\text{Re}\mathbf{X}}{\text{Im}\mathbf{X}}\right]$ is a vector containing the real part of $\mathbf{X}$ in the upper half and the imaginary part of $\mathbf{X}$ in the lower half.

Let us denote

$$\begin{aligned} \mathbf{A} &= \left[\frac{\mathrm{ReX}}{\mathrm{ImX}}\right] = \left[\frac{\mathbf{U}}{\mathbf{V}}\right] \\ \mathbf{B} &= \left[\frac{\mathrm{ReY}}{\mathrm{ImY}}\right] \\ \mathbf{C} &= \left[\frac{\mathrm{ReZ}}{\mathrm{ImZ}}\right] \end{aligned} \tag{3.13}$$

Note that the vector $\mathbf{A}$, of length N, is constructed such that lower half values are the real parts of X_k; $k = 1 \ldots N/2$ and the upper half is the imaginary part of X_k; $k = 1 \ldots N/2$. Similarly, $\mathbf{B}$ and $\mathbf{C}$, each of length N, contain real and imaginary parts of the lower half of the second-order correction and the observed Fourier components, respectively. The elements of $\mathbf{A}$ and $\mathbf{B}$ are denoted by a_l and b_k, respectively, where $l, k = 1 \ldots N$. The objective function in vector notation now is

$$\mathbf{f}(\mathbf{A}) = \mathbf{A} + \mathbf{B} - \mathbf{C} \tag{3.14}$$

A first-order Taylor approximation of $\mathbf{f}(\mathbf{A})$ about a given $\mathbf{A}^{(0)}$ is

$$\mathbf{f}(\mathbf{A}) = \mathbf{f}\left(\mathbf{A}^{(0)}\right) + [J]\left(\mathbf{A} - \mathbf{A}^{(0)}\right) \tag{3.15}$$

where $[J]$ is a $N \times N$ Jacobian matrix denoting the derivatives of the elements f_k in vector $\mathbf{f}(\mathbf{A})$ with respect to each of the unknowns a_l in $\mathbf{A}$ where $k, l = 1 \ldots N$. The Newton-Raphson scheme at iteration $p+1$ is then formulated as

$$\mathbf{A}^{(p+1)} = \mathbf{A}^{(p)} + \mathbf{h} \tag{3.16}$$

where $\mathbf{h}$, a vector of length N, is found from a Cholesky decomposition followed by a back-substitution scheme from

$$[J]\mathbf{h} = -\mathbf{f}\left(\mathbf{A}^{(p)}\right) \tag{3.17}$$

It can be easily shown from Eq. 3.14 that the entries $J_{k,l}$ of the matrix $[J]$ are

$$J_{k,l} = \frac{\partial f_k}{\partial a_l} = \delta_{kl} + \frac{\partial b_k}{\partial a_l} \tag{3.18}$$

where $\partial b_k / \partial a_l$ indicates the partial derivative of b_k with respect to a_l, and

$$\delta_{kl} = \begin{cases} 1 & \text{if } k = l \\ 0 & \text{otherwise} \end{cases} \tag{3.19}$$

To find $\partial b_k / \partial a_l$, recall from notation in 3.13

$$\begin{aligned} b_k = \mathrm{Re}Y_k \quad &\text{and} \quad b_{k+N/2} = \mathrm{Im}Y_k \quad \text{for } k = 1 \ldots N/2 \\ a_l = \mathrm{Im}X_l = U_l \quad &\text{and} \quad a_{l+N/2} = \mathrm{Im}X_l = V_l \quad \text{for } l = 1 \ldots N/2 \end{aligned}$$

so that from Eq.s 3.4 and 3.6 we have

$$\begin{aligned}
\frac{\partial \mathrm{Re}Y_k}{\partial U_l} &= \sum_{m+n,k} \left(U_n\delta_{ml} + U_m\delta_{nl}\right) H^+_{mn} + \sum_{m-n,k} \left(U_n\delta_{ml} + U_m\delta_{nl}\right) H^-_{mn} \\
\frac{\partial \mathrm{Re}Y_k}{\partial V_l} &= \sum_{m+n,k} -\left(V_n\delta_{ml} + V_m\delta_{nl}\right) H^+_{mn} + \sum_{m-n,k} \left(V_n\delta_{ml} + V_m\delta_{nl}\right) H^-_{mn} \\
\frac{\partial \mathrm{Im}Y_k}{\partial U_l} &= \sum_{m+n,k} \left(V_m\delta_{nl} + V_n\delta_{ml}\right) H^+_{mn} + \sum_{m-n,k} \left(V_m\delta_{nl} - V_n\delta_{ml}\right) H^-_{mn} \\
\frac{\partial \mathrm{Im}Y_k}{\partial V_l} &= \sum_{m+n,k} \left(U_n\delta_{ml} + U_m\delta_{nl}\right) H^+_{mn} + \sum_{m-n,k} \left(U_n\delta_{ml} - U_m\delta_{nl}\right) H^-_{mn}
\end{aligned} \tag{3.20}$$

Schematically,

$$[J] = [I] + \left[\begin{array}{c|c} \frac{\partial \mathrm{Re}Y_k}{\partial U_l} & \frac{\partial \mathrm{Re}Y_k}{\partial V_l} \\ \hline \frac{\partial \mathrm{Im}Y_k}{\partial U_l} & \frac{\partial \mathrm{Im}Y_k}{\partial V_l} \end{array}\right] \tag{3.21}$$

where $[I]$ is the identity matrix.

3.2.1 Newton-Raphson Scheme

The algorithm for the Newton-Raphson scheme followed in WAVEMAKER is

1. Estimate **C** from observed history (Eq.s 3.9, 3.13)
2. Initial Guess $\mathbf{A} = \mathbf{C}$
3. Estimate **B** from **A** (Eq.s 3.4, 3.6, 3.13)
4. Find $\mathbf{f}(\mathbf{A})$ (Eq. 3.14)
5. Find $[J]$ (Eq.s 3.20, 3.21)
6. Solve $[J]\mathrm{h} = -\mathbf{f}(\mathbf{A})$ to find **h**
7. Update $\mathbf{A} = \mathbf{A} + \mathbf{h}$
8. Check Convergence (see next section):
   If converged terminate else go to 3

3.2.2 Convergence Criteria

The Newton-Raphson iteration scheme is terminated based on the following conditions:

- **Program Converged**: If the rms of the increment vector $\mathbf{h} = \sqrt{\sum_1^N h_k^2/N}$ is less than a specified tolerance, the program is said to have converged. This convergence tolerance is specified as a fraction α (= 0.0001 in WAVEMAKER) of the standard deviation of the observed wave history $\sigma_{\eta,\text{obs}}$.
- **Program Diverging**: If the rms of the identified first-order history $\sigma_{\eta,1}$ at any iteration p is larger than a specified fraction β (= 200 in WAVEMAKER) of $\sigma_{\eta,\text{obs}}$ then the identification scheme is restarted with a smaller initial guess which is a truncated and scaled down version of $\mathbf{C}$. The truncation point is at twice the peak spectral frequency of $\eta_{\text{obs}}(t)$ and the scaling factor is $factnu^r$ (= 0.9^r in WAVEMAKER), where r is the number of restarts needed so far. Thus the restart guess in complex Fourier notation is

$$X_k = \begin{cases} factnu^r Z_k & ; \ \omega_k < 2\omega_{\text{peak}} \\ 0 & ; \ \text{otherwise} \end{cases} \tag{3.22}$$

- **Maximum Iterations Reached**: If the maximum allowed iterations, specified by the variable *mxiter* (= 10 in WAVEMAKER), is reached and the program has still not converged, then the program restarts the Newton-Raphson identification scheme with a smaller initial guess = $factnu^r\mathbf{C}$.
- **Maximum Restarts Reached**: If the maximum allowed restarts, specified by the variable *nuiter* (= 5 in WAVEMAKER), is reached then the program terminates the identification scheme in the present window and proceeds to identify in the next observed history window.

3.2.3 Implementation

The first-order components for the observed wave history, of length N_{obs}, are identified in contiguous windows, each of length $N \leq N_{\text{obs}}$. The identification analysis is performed in this way to minimize the computer memory usage by WAVEMAKER. Recall that the Jacobian matrix $[J]$ is a $N \times N$ matrix and the memory usage is directly governed by the matrix size of this variable. In principle, if there is sufficient memory we could set $N = N_{\text{obs}}$ and identify the first-order component for the entire observed history in one window, however, this is not usually not the case and we resort to identifying in contiguous windows, as shown in Fig. 3.1.

The first-order component is identified independently in each of the windows in sequence. The last window is skipped if its contains points less than N.

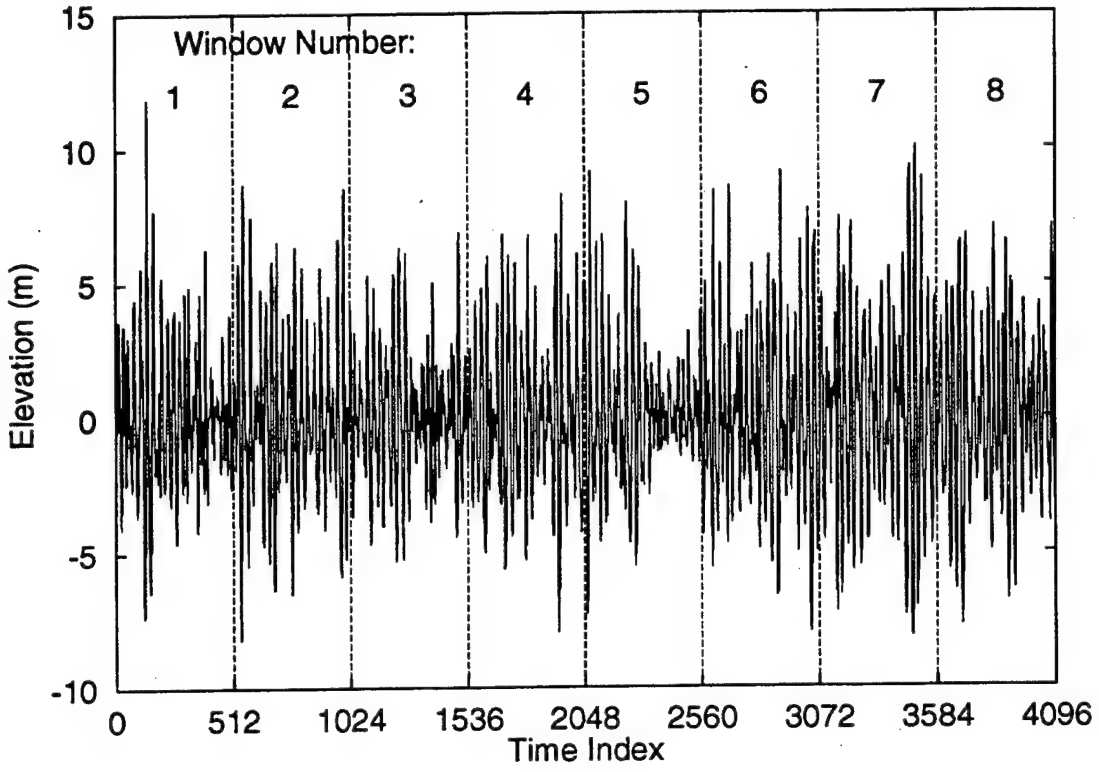

Figure 3.1: Identification of first-order wave components is done in contiguous windows of the observed history

3.3 Input Specification

The input specification for the identification of first-order wave process is in a command-line format similar to the simulation input. A typical input file for identification is:

```
# Typical input file: syntax description

identify filename dt winsize
depth value
define varlimit value
define gravity value
write history filename1 filename2
```

 # Typical input file: syntax description
Any line beginning with a "#" symbol is treated as a comment line and is ignored.

Blank lines in the input file are ignored, as well.

define varlimit *value*
define gravity *value*
The keywords **varlimit**, **gravity** have the same meaning as in the simulation section and the user is referred to this section to understand the usage of these commands.

depth *value*
The keyword **depth** as in the simulation section indicates the water depth at which the identification analysis is to be performed.

identify *filename dt winsize*
The keyword **identify** indicates to the program that the user intends to identify the underlying first-order wave history for a given observed wave history. This command requires three arguments which in sequence are:

- *filename*: The name of the file, a character string, containing the observed wave time history for which the underlying first-order wave history is to be identified. The data in the **first** column in *filename* is read as the observed wave time history. Any blank lines in *filename* or lines that do not begin with a number are ignored.
- *dt*: This value, a real number, indicates the time resolution of the wave history provided in *filename*. In other words, *dt* is the time difference between two successive elevation values in the observed wave history.
- *winsize*: An integer value indicating the window size or the number of points of the provided wave history to be used in each Newton-Raphson iteration. The first-order wave components are identified in windows (of size *winsize*) in sequence for the provided time history. If the last window contains number of points less than *winsize* then this window is ignored and the first-order components are not identified in this window.

The maximum value of *winsize* is *mlen* set to 512 points in `WAVEMAKER` and can be changed according to the user's needs or according to the computer's limitations. Note that we require $mlen < 2 \times mxgrd$ in the program. These dimension values are set in this way so as to minimize the memory requirements of `WAVEMAKER`.

write history *filename1 filename2*
The command **write history** is used to specify the file names where the identified histories are to be written. The identified first-order wave history is written in file *filename1* and the identified second-order, combined first- and second-order, and the observed wave histories are written in file *filename2*. Default values assigned to *filename1* and *filename2* are **gauss.hist** and **ngauss.hist**, respectively.

3.4 Output Format

The output file names are governed by the command
write history *filename1 filename2*
with default names being **gauss.hist** and **ngauss.hist** for *filename1* and *filename2*, respectively.

filename1 contains the identified first-order wave history in a column of real numbers. Each line of this file contains one real number indicating the elevation of the first-order wave history (see example output files in Appendix B). The time resolution of this first-order history is *dt*, equal to the *dt* provided in the input file using the command **identify**. This file also contains comment lines that begin with a "#" symbol as the first character on the line. The first comment line contains information on the contents of the file, and the following comment lines contain 3 integers: the first is the window number being identified, the second is the number of iterations required for convergence, and the third is the number of restarts needed for convergence.

filename2 contains the second-order correction, the combined first- and second-order waves, and the observed wave time history. The second-order correction is found from the identified first-order waves, and these two are added together to yield the combined second-order wave history. These histories are provided in three columns in *filename2*, or in other words each line contains three real numbers: the first is the second-order wave elevation, the second is the combined identified wave elevation, and the third is the observed wave history (see example output file in Appendix B). A match of the total identified and the observed wave histories will verify successfuly identification by WAVEMAKER. The time resolution of each of these histories is *dt*. This file also contains comment lines beginning with a "#" symbol that provides information similar to the comment lines in *filename1*.

3.5 Examples

In this section we present two sample problems to illustrate the use of the identification capabilities of WAVEMAKER. Example 1 is based on the example presented in the simulation chapter. Sample input and output files of this identification example are included in the distribution diskette. Example 2 presented here demonstrates the identification of first-order components of a measured wave tank history. Note that sample input or output files of this second example are not included in the distribution.

3.5.1 Example 1

The example of the simulation capabilities of **WAVEMAKER** involved simulating a second-order wave history characterized by a JONSWAP spectrum with $H_s = 12$ [m], $T_p = 14$s and $\gamma = 3.3$ in 70 [m] water depth. We will use the combined second-order simulated history and try to identify its first-order wave component and compare it to the input first-order component used to simulate the combined wave history. The input file for the identification run is

```
# Wave Identification Input File

identify hist.dat 0.5 512
depth 70.0
write history gauss.ide ngauss.ide
define varlimit 0.01
define gravity 9.807
```

Alternatively, if we intend to use the default definitions in the program then our input file could be (this will produce the same output as the extended version of the input file):

```
# Wave Identification Input File
# (Alternative format)

identify hist.dat 0.5 512
depth 70.0
write history gauss.ide ngauss.ide
```

The input file **hist.dat** contains a column of real numbers (see sample files listed in the appendix) which will be read in as the observed wave history. The first-order components will be identified for this observed history and placed in the file **gauss.ide**. The corresponding second-order components, the combined identified history and the observed wave history are written in the file **ngauss.ide**.

Output File	Contents
gauss.ide	Identified First-Order Time History
ngauss.ide	Corresponding Second-Order Time History

Figure 3.2 shows the observed wave spectrum and the identified first-order spectrum along with the corresponding second-order wave spectrum. We see that small second-order contribution to the power spectrum, roughly a decade below the first-order spectrum even at frequencies twice the peak spectral frequency, suggests the difficulty in identifying these components. Figure 3.3 shows the observed wave history and the identified first-order wave history in cycles around the maximum crest height. Compare this to the simulation example where we solve the forward problem of finding the combined (first- plus second-order) history from a given underlying first-order wave spectrum. The identified first-order component in Fig. 3.3 is almost the same as the underlying first-order component (denoted Gaussian) in Fig. 2.2 and these two are shown together in Fig. 3.4. Note how close the two first-order components are, and any numerical differences can probably be further reduced by using a larger window size (greater than 512, for example) in the identification scheme.

3.5.2 Example 2

In this example we will identify the underlying first-order wave component for a measured wave tank history that reflects a water depth of about 300m. For this example the measured history is located in file **wave.dat** and has a $dt = 0.424264$ seconds. We will use windows of $winsize = 512$ to identify the first-order components. The input to `WAVEMAKER` is:

```
# Wave Identification Input File
# using default definitions

identify wave.dat 0.424264 512
depth 300.0
```

Figure 3.5 shows a portion where the maximum crest height occurs in the measured wave tank history. The figure also shows the identified first-order and the corresponding second-order wave histories. Note how the second-order wave component affects the first-order peaks, amplifying the crests and moderating the troughs. Figure 3.6 shows the wave spectra for the measured history along with the first-order and the second-order spectra. Again, observe that the second-order energy is significantly small compared to the first-order, however, phase locking of the first- and the second- component (Fig. 3.5) leads to larger crests and flatter troughs.

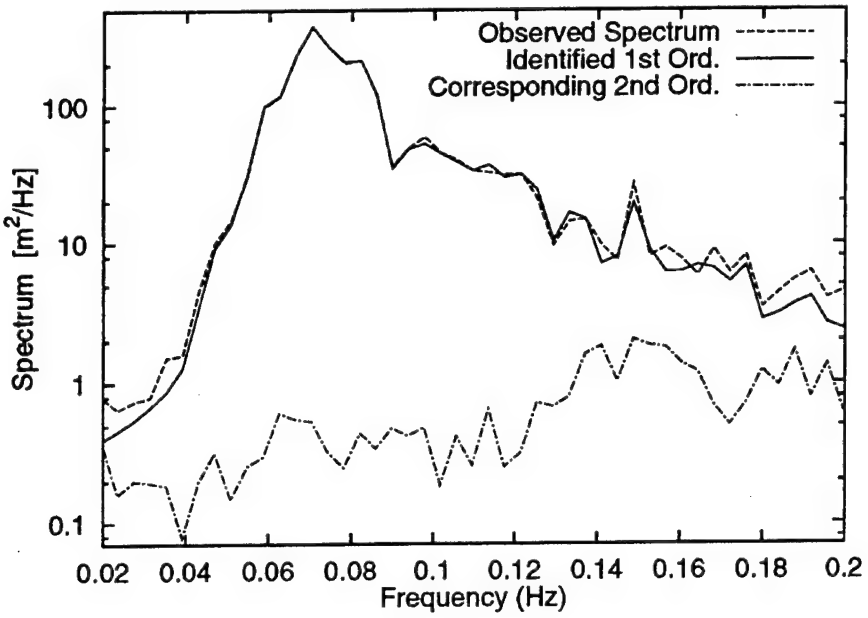

Figure 3.2: Wave spectrum: observed vs. identified first- and second-order

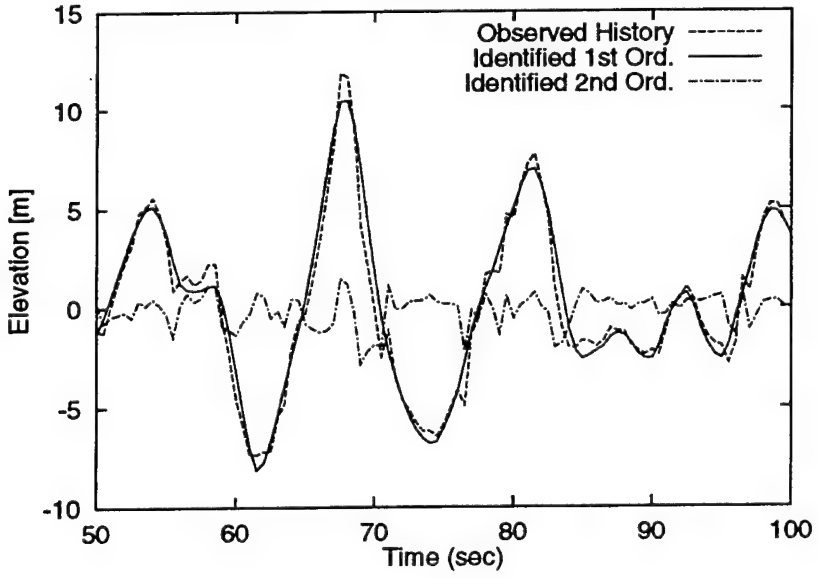

Figure 3.3: Wave history: observed vs. identified first- and second-order

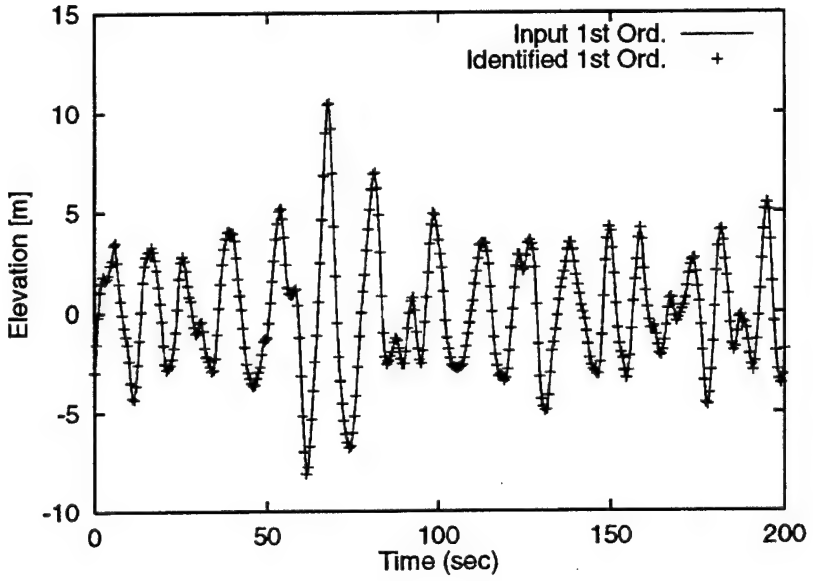

Figure 3.4: Identified first-order vs. actual first-order wave history

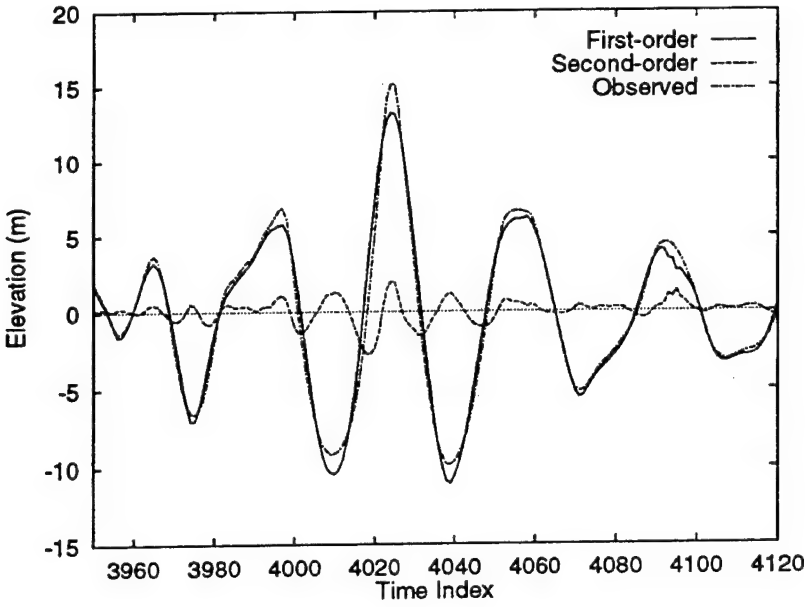

Figure 3.5: Wave history in wave tank: observed vs. identified first- and second-order

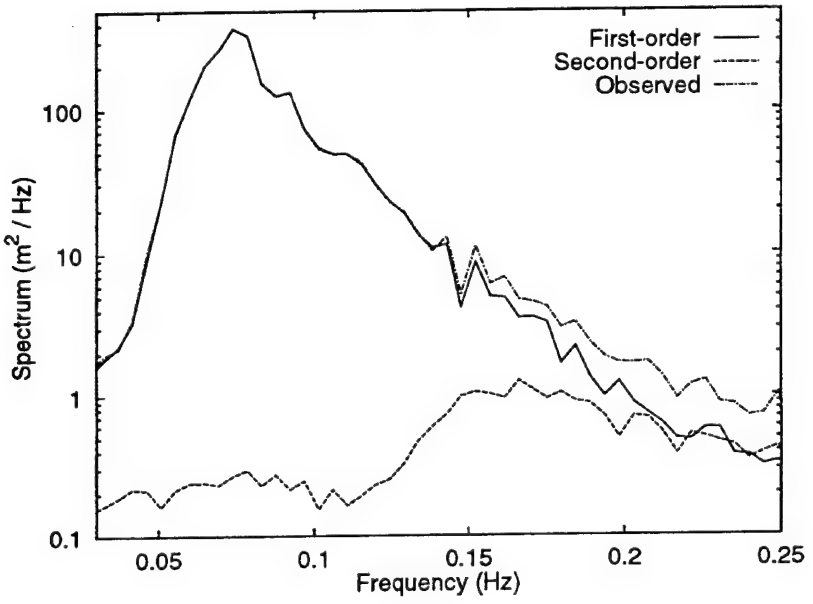

Figure 3.6: Wave spectrum in wave tank: observed vs. identified first- and second-order

Chapter 4

Distribution

The WAVEMAKER routine and example files have been distributed on a DOS formatted 3.5 inch floppy diskette. The diskette contains the source code files (of the form `*.f`), example input (`*.inp`), and output files (`*.sta`, `*.his`, and `*.ide`). The diskette also contains this manual in postscript format in `manual.ps`.

4.1 Copying the Diskette

Copy the contents of the diskette on to your host computer (computer on which you will run WAVEMAKER). After the copying is done, your host computer should have:

- Example input files `wavmkr.inp`, `wavide.inp`, and `hist.dat`.
- Example output files `gauss.his`, `ngauss.his`, `gauss.sta`, `ngauss.sta`, `gauss.ide`, and `ngauss.ide`.
- `manual.ps` containing this manual in postscript format
- all the source files `*.f` and `Makefile`

Table 4.1 shows the input and output files that are specific to the simulation and to the identification examples.

Table 4.1: Distributed Files for Simulation and Identification Examples

Simulation Example		
File Type	**File Name**	**Description**
Input File	`wavmkr.inp`	Input to WAVEMAKER
Output Files	`gauss.his`	Simulated first-order wave histories at each specified location
	`ngauss.his`	Simulated combined first- and second-order wave histories at each specified location
	`gauss.sta`	Statistics of the simulated first-order histories at each specified location
	`ngauss.sta`	Statistics of the simulated combined histories at each specified location

Identification Example		
File Type	**File Name**	**Description**
Input Files	`wavide.inp`	Input to WAVEMAKER
	`hist.dat`	Observed wave history for which underlying first-order history is to be identified
Output Files	`gauss.ide`	Identified first-order wave history
	`ngauss.ide`	Identified second-order, combined first- and second-order, and observed wave histories

4.2 Compiling the Source

On a Unix workstation, the `Makefile` can be used to build the WAVEMAKER executable. To compile on your host Computer:

- change directory to the subdirectory containing the source code files
- type "make wavmkr" (without the double quotes) at the Unix prompt and press return

The above will compile all the files listed in `Makefile` and link them to make the executable `wavmkr`. Note that the executable `wavmkr` is still in the current directory, and may be moved to the directory which contains the input file, or you can specify the path to the executable file in order to run it from other directories.

On other operating systems and architectures, follow whatever is the standard procedure for compiling and linking FORTRAN source code distributed over multiple files.

WAVEMAKER has been developed on a Sun Sparcstation2, using version 1.4 of the Sun FORTRAN compiler. Every attempt has been made to adhere to the ANSI FORTRAN 77 standard, ensuring portability of the code.

4.3 Executing the Routine

At the Unix prompt type "wavmkr < wavmkr.inp" (without the double quotes) in order to execute WAVEMAKER and perform the simulation example analysis. The program reads input from the standard logical input unit. The logical units for standard input, standard output, and standard error are all used in WAVEMAKER. On the Sun compiler, 0 is used for standard error, 5 is standard input and 6 is standard output. If the appropriate unit numbers are different, they can be set using the IOER, IOIN, and IOOU variables in the WAVEMAKER driver program, and the package can be recompiled.

Run the example problem using the complied code to check if you get the same output as provided in the example output files. Note that wavmkr will overwrite any existing file if you specify its name as the target output file using the `write` command in the input file.

Similarly, at the Unix prompt type "wavmkr < wavide.inp" (without the double quotes) in order to execute WAVEMAKER and perform the identification example analysis. The resulting output histories can be compared to the corresponding example output histories to verify successful compilation of WAVEMAKER.

List of References

Bitner-Gregerson, E., and Haver, S. 1991. Joint environmental model for reliability calculations. *Pages 472–478 of: Proceedings of the 1st international offshore and polar engineering conference (ISOPE).*

Borgman, L. E. 1969. Ocean wave simulation for engineering design. *Journal of waterways, harbors division*, **95**(4), 556–583. ASCE.

Jha, A. K., and Winterstein, S. R. 1995. *Second-order random waves: Simulation in time and space.* Tech. rept. RMS-17. Reliability of Marine Structures Program, Stanford University, Dept. of Civil Engineering.

Langley, R. S. 1987. A statistical analysis of non-linear random waves. *Ocean engineering (pergamon)*, **14**(5), 389–407.

Longuet-Higgins, M. S. 1963. The effect of non-linearities on statistical distributions in the theory of sea waves. *Journal of fluid mechanics*, **17**(3), 459–480.

Marthinsen, T., and Winterstein, S. R. 1992. On the skewness of random surface waves. *Pages 472–478 of: Proceedings of the 2nd international offshore and polar engineering conference, san francisco.* ISOPE.

Molin, B, and Chen, X. B. 1990. *Calculation of second-order sum-frequency loads on tlp hulls.* Tech. rept. Institute Français du Petrele Report.

SWIM, 2.1. 1995. *SWIM: Slow wave-induced motions– user's manual.* Dept. of Ocean Engineering, M.I.T.

Ude, T. C. 1994. *Second-order load and response models for floating structures: Probabilistics analysis and system identification.* Tech. rept. RMS–16. Reliability of Marine Structures, Dept. of Civil Engr., Stanford University.

Ude, T. C., and Winterstein, S. R. 1996. Calibration of models for slow drift motions using statistical moments of observed data. *In: 6th international offshore and polar engineering conference ISOPE.* to appear.

Vinje, T., and Haver, S. 1994. On the non-Gaussian structure of ocean waves. *Pages 453–480 of: Proceedings of the 7th international conference on the behaviour of offshore structures (BOSS)*, vol. 2.

WAMIT, 4.0. 1995. *WAMIT: A radiation-diffraction panel program for wave-body interactions–user's manual.* Dept. of Ocean Engineering, M.I.T.

Winterstein, S. R., and Jha, A. K. 1995. Random models of second-order waves and local wave statistics. *Pages 1171–1174 of: Proceedings of the 10th engineering mechanics speciality conference.* ASCE.

Appendix A

Output Files for Simulation Example

Output File: **gauss.his**

```
# Underlying First-Order Wave Process
#                Wave Elevation at Spatial Location
#  Time(sec.)      0.00     60.00
      0.000000   -3.189    -4.754
      0.500000   -1.754    -5.466
      1.000000   -0.239    -5.410
      1.500000    0.906    -4.667
      2.000000    1.436    -3.660
      2.500000    1.681    -2.714
      3.000000    1.623    -1.676
      3.500000    1.595    -0.875
      4.000000    1.862    -0.137
      4.500000    2.282     0.582
      5.000000    2.802     1.211
      5.500000    3.230     1.379
      6.000000    3.263     1.263
      6.500000    2.448     1.252
      7.000000    1.403     1.572
      7.500000    0.553     2.166
      8.000000   -0.121     2.447
      8.500000   -0.773     3.076
      9.000000   -1.216     3.472
      9.500000   -1.588     3.272
     10.000000   -2.433     2.507
     10.500000   -3.543     1.492
     11.000000   -4.310     0.022
     11.500000   -4.343    -1.441
         :          :         :
         :          :         :
         :          :         :
   2046.500000   -3.871     0.056
   2047.000000   -4.266    -1.729
   2047.500000   -4.040    -3.447
```

Note that ":" indicates more numbers. Since the files are long, we present truncated versions of the output files.

Output File: **ngauss.his**

```
# Total Second-Order Wave Process
#                Wave Elevation at Spatial Location
#  Time(sec.)      0.00     60.00
     0.000000    -3.487    -4.814
     0.500000    -2.747    -5.039
     1.000000    -0.970    -4.773
     1.500000     0.975    -4.589
     2.000000     1.927    -3.459
     2.500000     1.920    -2.768
     3.000000     2.210    -1.844
     3.500000     1.738    -0.534
     4.000000     2.034    -0.256
     4.500000     2.353     0.534
     5.000000     2.974     1.072
     5.500000     3.595     2.033
     6.000000     3.481     1.523
     6.500000     1.913     1.684
     7.000000     1.236     1.334
     7.500000     0.644     2.374
     8.000000     0.019     2.724
     8.500000    -0.826     3.160
     9.000000    -0.572     3.945
     9.500000    -1.304     3.330
    10.000000    -2.734     2.573
    10.500000    -3.700     1.296
    11.000000    -3.854    -1.195
    11.500000    -4.087    -1.811
    12.000000    -3.686    -2.346
    12.500000    -2.784    -2.356
    13.000000    -1.532    -2.498
    13.500000    -1.026    -2.256
        :            :         :
        :            :         :
        :            :         :
  2046.500000    -3.594    -1.165
  2047.000000    -3.918    -3.159
  2047.500000    -3.660    -4.297
```

Output File: **gauss.sta**

```
# Underlying First-Order Wave Process
# Location      Mean      Sigma      Skewness    Kurtosis    Minimum      Maximum
    0.00   0.9966E-09   2.975      0.3566E-01   2.917      -8.928        10.41
   60.00   0.1000E-08   2.975      0.2010E-01   2.919      -9.241        9.784
```

Output File: **ngauss.sta**

```
# Total Second-Order Wave Process
# Location      Mean      Sigma      Skewness    Kurtosis    Minimum      Maximum
    0.00   -.3673E-09   3.027      0.1949       2.975      -8.235        11.85
   60.00   -.1193E-08   3.037      0.1778       2.988      -8.576        11.18
```

Appendix B

Input/Output Files for Identification Example

Input File: **hist.dat**

```
-3.487
-2.747
-0.970
 0.975
 1.927
 1.920
 2.210
 1.738
 2.034
 2.353
 2.974
 3.595
 3.480
 1.914
 1.236
 0.644
 0.019
-0.826
-0.572
-1.304
-2.734
-3.700
   :
   :
   :
-1.458
-2.043
-2.767
-3.241
-3.594
-3.918
-3.660
```

Note that ":" indicates more numbers. Since the files are long, we present truncated versions of the output files.

Output File: **gauss.ide**

```
# Indentified First-Order Wave History
# Window:  number   iterations    restarts
# Window      1     6     0
   -2.9780
   -1.5528
   -0.1932
    1.0374
    1.4503
       :
   -0.4398
    0.9539
   -1.0007
# Window      2     8     0
    0.6255
    0.8085
    0.5614
    0.5256
    1.0275
    0.9476
    1.1145
       :
       :
# Window      8     5     0
   -2.2065
   -1.4332
   -1.1666
   -1.0438
   -0.8492
       :
   -3.2288
   -3.7402
   -4.1472
   -3.4963
```

Output File: ngauss.ide

```
# Identified 2nd-Order Wave, Identified Total Wave, Input Wave History
# Window:  number   iterations    restarts
# Window      1     6     0
   -0.5090   -3.4870   -3.4870
   -1.1942   -2.7470   -2.7470
   -0.7768   -0.9700   -0.9700
   -0.0624    0.9750    0.9750
    0.4767    1.9270    1.9270
       :         :         :
   -0.0489   -1.7070   -1.7070
   -0.8702   -1.3100   -1.3100
   -0.3039    0.6500    0.6500
    2.0597    1.0590    1.0590
# Window      2     8     0
    0.1029    0.7284    0.7320
    0.4542    1.2627    1.2660
    0.3126    0.8740    0.8710
    0.0864    0.6120    0.6120
    0.1123    1.1399    1.1390
    0.4375    1.3851    1.3860
    0.0240    1.1385    1.1380
       :         :         :
       :         :         :
# Window      8     5     0
   -0.3825   -2.5890   -2.5890
    0.3802   -1.0530   -1.0530
    0.3296   -0.8370   -0.8370
    0.2628   -0.7810   -0.7810
    0.1202   -0.7290   -0.7290
       :         :         :
   -0.0122   -3.2410   -3.2410
    0.1462   -3.5940   -3.5940
    0.2292   -3.9180   -3.9180
   -0.1637   -3.6600   -3.6600
```

Appendix C

Random Models of Second-Order Waves and Local Wave Statistics

The following paper demonstrates the application of these nonlinear wave models and simulation techniques. It also shows how wave moments can be estimated analytically, and resulting estimates of extreme waves formed. Finally, various local wave characteristics found from the simulation are compared with field and wave tank data.

It has appeared in the Proceedings of the 10th Engineering Mechanics Specialty Conference, ASCE, held in Boulder, May 1995.

From *Proceedings*, 10th ASCE Eng. Mech. Specialty Conference
Boulder, Colorado, Vol. 2, 1995, pp. 1171–1174

Random Models of Second-Order Waves and Local Wave Statistics

Steven R. Winterstein and Alok K. Jha
Civil Eng. Dept., Stanford University

Abstract

We consider second-order random models of ocean waves at arbitrary water depths. We derive convenient new analytical results for wave moments, and show results for crests and other local wave statistics. Theoretical predictions are compared with observed wave tank results in extreme seas.

Introduction

Nonlinear hydrodynamic effects are of growing interest for ocean structures and vessels. This has spurred development of efficient methods to estimate statistics of second-order hydrodynamic models (e.g., Winterstein et al, 1994). Here we apply these to one of the most fundamental nonlinearities in ocean engineering: the wave elevation $\eta(t)$ at a fixed spatial location.

Linear wave theory results in a Gaussian model of $\eta(t)$. This ignores the marked asymmetry of $\eta(t)$: wave crests that systematically exceed subsequent troughs. This has several practical implications: (1) asymmetric waves are more likely to strike decks of offshore platforms, particularly older Gulf-of-Mexico structures with fairly low decks; and (2) unusually large dynamic response has been found in high, steep waves that may not follow linear theory.

Second-order random wave models are not new; indeed, they have been a research topic for more than 30 years (e.g., Longuet-Higgins, 1963) and remain so today (e.g., Marthinsen and Winterstein, 1992; Hu and Zhao, 1993; Vinje and Haver, 1994). However, they have not yet entered common offshore engineering practice, which applies either (1) random linear (Gaussian) waves or (2) regular Stokes waves that fail to preserve $S_\eta(\omega)$, the wave power spectrum. Several drawbacks to second-order random waves may be suggested: (1) they omit potentially important higher-order effects; and (2) convenient statistical analysis methods for second-order models are often lacking. We seek to address both concerns here—the first through systematic comparison of theory with observed wave tank results in extreme seas. The second issue is met by fitting new analytical results for wave moments, and using these to construct simple Hermite models of extreme crests.

Statistics of Second-Order Models

Our "input" is the first-order, Gaussian wave process $\eta_1(t)$ from linear theory. The standard Fourier sum for $\eta_1(t)$ is then $\mathrm{Re}\sum C_k \exp(\mathrm{i}\omega_k t)$, in which $C_k{=}A_k \exp(\mathrm{i}\phi_k)$ in

terms of Rayleigh distributed amplitudes, A_k, and uniformly distributed phases ϕ_k. The resulting "output" is $x(t)=x_1(t)+x_2(t)$, in which

$$x_1 = \mathrm{Re}\sum_k C_k H_k e^{i\omega_k t};\quad x_2 = \mathrm{Re}\sum_k\sum_l C_k C_l [H_{kl}^{+} e^{i(\omega_k+\omega_l)t} + H_{kl}^{-} e^{i(\omega_k-\omega_l)t}] \qquad (1)$$

Here the transfer function H_k describes first-order effects, while H_{kl}^{+} and H_{kl}^{-} reflect second-order effects at sums and differences of all wave frequencies $(\omega_k \pm \omega_l)$. In our case $x(t)$ is the second-order wave itself, for which H_{kl}^{+} and H_{kl}^{-} are given analytically (e.g., Marthinsen and Winterstein, 1992), and H_k=1. The same analysis applies more generally to the diffracted wave, applied force and response of large-volume structures, with numerical H_k and H_{kl} estimates from second-order diffraction (Winterstein et al, 1994).

Because $x(t)$ is non-Gaussian, interest focuses on its skewness α_3 and kurtosis α_4. In terms of the significant wave height $H_S=4\sigma_{\eta_1}$ and peak spectral period T_P, these are

$$\alpha_3\sigma_x^3 = m_{31}(T_P)H_S^4 + m_{33}(T_P)H_S^6 \qquad (2)$$

$$(\alpha_4 - 3)\sigma_x^4 = m_{42}(T_P)H_S^6 + m_{44}(T_P)H_S^8 \qquad (3)$$

The $m_{ij}(T_P)$ are "response moment influence coefficients," the contribution to response moment (cumulant) i due to terms of order $O(x_2^j)$. In general these are conveniently calculated from Kac-Siegert analysis (Eqs. 12–15, Winterstein et al, 1994). We assume here the spectrum of $\eta_1(t)$ is of the form $H_S^2 T_P f(\omega T_P)$, so that $\eta_1(t)$ scales in amplitude with H_S and in time with T_P.

It is useful to define the unitless wave steepness $s_P=H_S/L_P$, in which the characteristic wave length $L_P=gT_P^2/2\pi$ uses the linear dispersion relation. For deep-water waves the coefficients $m_{ij}(T_P)$ are proportional to L_P^{-j}. Retaining leading terms in s_P from Eqs. 2–3:

$$\alpha_3 = k_3 s_P;\quad \alpha_4 - 3 = k_4\alpha_3^2 \qquad (4)$$

In particular, for a JONSWAP wave spectrum with peakedness factor γ, we have fit the following k_3 and k_4 expressions to results for a wide range of depths:

$$k_3 = \frac{\alpha_3}{s_P} = 5.45\gamma^{-.084} + .135\left(\frac{d}{L_P}\right)^{-1.22};\quad k_4 = \frac{\alpha_4 - 3}{\alpha_3^2} = 1.41\gamma^{-.020} \qquad (5)$$

The second term in this result for α_3 reflects the effect of a finite water depth d: in shallower water α_3 grows, as the waves begin to "feel" the bottom.

Note also that while the skewness varies linearly with steepness, the kurtosis varies quadratically. This suggests that nonlinear effects will be most strongly displayed by the skewness, and hence by the wave crests rather than the total peak-to-trough wave height. This second-order model may less accurately predict kurtosis, however, as higher-order omitted effects may be of the same order of magnitude.

Numerical Results

Figure 1 compares skewness predictions with results from wave tank tests. The tests include 18 large seastates (target H_S=14.5m–15.5m), each over 3 hours in length, at water depths exceeding 300m. Figure 1 shows the resulting skewness and steepness in each hour of each test. While there is considerable scatter, regression on these data

gives the estimated slope k_3=5.50, remarkably close to Eq. 5 with γ=1. The scatter in Figure 1 is also consistent: the observed σ_{α_3} is found well-predicted by simulated hourly segments of second-order seastates. Figure 2 shows kurtosis estimates to deviate, however. The data yield the estimate k_4=4.2, roughly 3 times the value in Eq. 5 regardless of γ. This again supports theory, which suggests that unlike the skewness, the kurtosis may be notably affected by unmodelled, higher-order effects.

Wave Crests. Figure 3 shows the observed distribution of crest heights. These results combine six seastates with the same target spectrum, and hence give roughly 20 hours of similar wave conditions. As expected the Rayleigh model, based on linear theory, is significantly unconservative. An alternative empirical model (Haring et al, 1976) offers an improvement, but only mildly changes the Rayleigh for these deep-water conditions. Better agreement is found from a Non-Gaussian (Hermite) model, which uses a cubic distortion of the normal process (and hence its Rayleigh peaks) to match α_3 and α_4 (Winterstein et al, 1994). Here estimates of α_3 and α_4 use Eq. 5, tripling its k_4 value to reflect unmodelled effects. These give excellent agreement with observed moments, though still somewhat unconservative crest predictions at higher levels.

Local Wave Characteristics. Figure 4 shows the conditional mean and standard deviation of crest height, given the peak-to-trough wave height. Again the data use 20 hours of wave tank studies, all with the same target wave spectrum. Also shown are corresponding estimates from simulation of Gaussian and Non-Gaussian (second-order) models. Due to the symmetry of the Gaussian model, its crests are on average 50% of the total wave height. The data shows systematically larger crests. The second-order model is found to predict this vertical asymmetry quite accurately.

We may also consider horizontal wave asymmetry: do crest front periods—during which η increases from its mean level to a crest—differ statistically from subsequent crest back periods? This temporal asymmetry is not predicted by either the Gaussian or second-order model. It is difficult, however, to find this asymmetry in the data: Figure 5 shows observed crest fronts to be quite close to 50% of the total crest period. Finally, Figure 6 shows the variation of wave period with crest height. All results show the same trend of increasing periods over small-to-moderate heights, and roughly constant period at large heights. The Non-Gaussian model appears to somewhat better predict results at larger crest height levels.

References

Haring, R.E., A.R. Osborne, and L.-P. Spencer (1976). Extreme wave parameters based on continental shelf storm wave records. *Proc., 15th Coastal Eng. Conf., ASCE,* 151–170.

Hu, S.J. and D. Zhao (1993). Non-Gaussian properties of second-order random waves. *J. Engrg. Mech.,* ASCE, **119**(2), 344–364.

Longuet-Higgins, M.S. (1963). The effect of non-linearities on statistical distributions in the theory of sea waves. *J. Fluid Mech.*, **17**(3), 459–480.

Marthinsen, T.M. and S.R. Winterstein (1992). On the skewness of random surface waves. *Proc., 2nd Intl. Offshore Polar Eng. Conf.,* ISOPE, 472–478.

Vinje, T. and S. Haver (1994). On the non-Gaussian structure of ocean waves. *Proc., BOSS-94,* **2**, 453–480.

Winterstein, S.R., T.C. Ude and T. Marthinsen (1994). Volterra models of ocean structures: extreme and fatigue reliability. *J. Engrg. Mech.,* ASCE, **120**(6), 1369–1385.

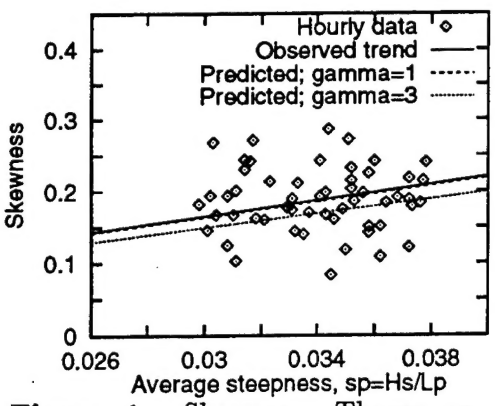

Figure 1: Skewness: Theory vs Data

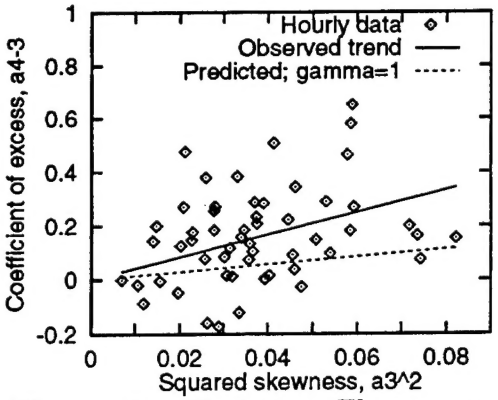

Figure 2: Kurtosis: Theory vs Data

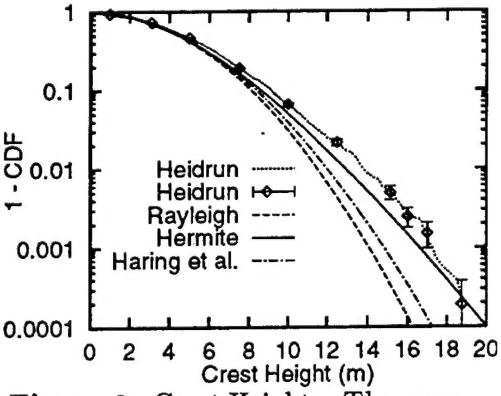

Figure 3: Crest Heights: Theory vs Data

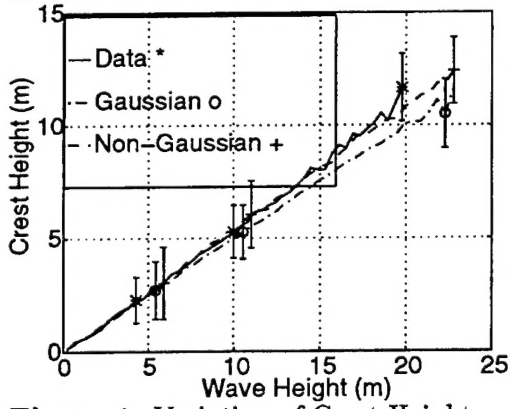

Figure 4: Variation of Crest Height with Wave Height

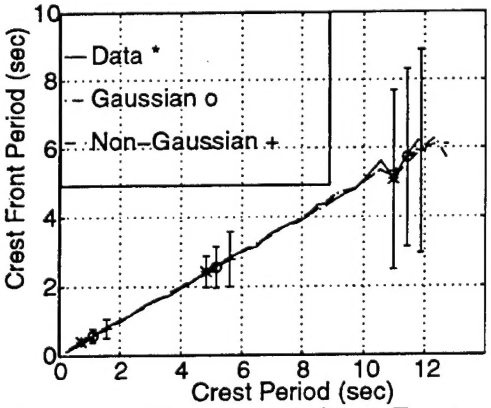

Figure 5: Variation of Crest Front Period with Total Crest Period

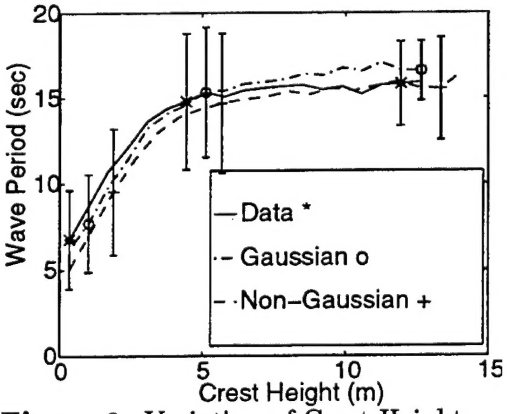

Figure 6: Variation of Crest Height with Wave Period

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. REPORT DATE (DD-MM-YYYY)	2. REPORT DATE	3. DATES COVERED (From - To)
00-06-1996	June 1996	

4. TITLE AND SUBTITLE
Wavemaker 20: Simulation and Identification of Second-Order Random Waves.

5a. CONTRACT NUMBER

5b. GRANT NUMBER
N00014-96-1-0641

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)
Alok K. Jha
Steven R. Winterstein

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
RMS Group
S.R. Winterstein, C.A. Cornell
Blume Center
Stanford University, CA 94305

8. PERFORMING ORGANIZATION REPORT NUMBER
RMS-22

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
Office of Naval Research
800 North Quincy St
Arlington, VA 22217-4620
ATTN: Roshdy Barsoum

10. SPONSOR/MONITOR'S ACRONYM(S)

11. SPONSORING/MONITORING AGENCY REPORT NUMBER

12. DISTRIBUTION AVAILABILITY STATEMENT
Approved For Public Release

13. SUPPLEMENTARY NOTES

14. ABSTRACT
This report documents the theory and use of the routine WAVEMAKER. The routine simulates a first-order wave process with an arbitrary power spectrum, and applies nonlinear corrections based on second-order hydrodynamics. The routine also includes the ability to identify the underlying first-order Gaussian history from a given observed time history. The identification is an inverse feature to simulation, and is based on a Newton-Raphson scheme to solve N simultaneous non-linear equations, where N is the number of specified points in the input time-history.

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			19b. TELEPONE NUMBER (*Include area code*)